

Estudio y uso de la tecnología de realidad aumentada en smartphones

Pere Garcia Sanahuja

Resumen—La realidad aumentada es un término referente a un sistema que permite combinar en tiempo de ejecución, la visión real del mundo con otra visión virtual generada mediante un dispositivo electrónico. Esta visión combinada se mostrará a través del dispositivo electrónico y permitirá al programador, alterar la visión real del mundo utilizando las tecnologías de visión por computador. El objetivo de este trabajo es experimentar con las distintas librerías de realidad aumentada que existen a día de hoy y establecer un primer contacto con el IDE UNITY. El resultado final es una demostración de los distintos elementos que componen una escena tridimensional, su interacción con la pantalla táctil del dispositivo y la librería de realidad aumentada.

Palabras clave—AR, Realidad Aumentada, PTAM, VSLAM, Android, Smartphone, Metaio, Vuforia, Qualcomm, UNITY, Assets, frame.

Abstract—The augmented reality is a system that combines the reality with a virtual scene, generated by Smartphone, in runtime. This combined view is displayed through the Smartphone screen and allow the device developer alter the actual worldview using computer vision technologies. The aim of this work is to experiment with different augmented reality libraries that exist today and establish a first contact with the IDE UNITY. The result is a demonstration of various components of a three-dimensional scene, their interaction with the touch screen of the device and the augmented reality library.

Keywords—AR, Augmented Reality, PTAM, VSLAM, Android, Smartphone, Metaio, Vuforia, Qualcomm, UNITY, Assets, frame.



1 INTRODUCCIÓN

La realidad aumentada es un sistema que permite superponer una realidad virtual generada por un dispositivo electrónico a la realidad física captada mediante un dispositivo de captura de imágenes.

El sistema se compone de una base de datos de imágenes. Cada una de las imágenes que componen esta base de datos se llama “image target” y tienen definidas características únicas como el cromatismo y la intensidad de la luz.

Este sistema se encarga de identificar en cada una de las imágenes los distintos puntos de correspondencia de los objetivos. Cuando el sistema detecta uno de los patrones de alguna de las imágenes de la base de datos utilizada, se procede a generar la realidad virtual asociada a ella.

Este sistema tiene mucha ventaja en dispositivos Smartp-

hone ya que estos tienen todos los componentes necesarios para ejecutarlo de forma óptima.

Además de este sistema, existen tecnologías alternativas que permiten generar la realidad virtual sin utilizar los patrones “image targets”.

El marco del trabajo realizado corresponde al estudio de las cualidades y defectos que tienen las librerías actuales del sistema AR ejecutadas bajo un dispositivo Smartphone. Todo el trabajo está enfocado bajo esta plataforma.

Se pondrán a prueba las librerías y el mismo Smartphone que generará el contenido. Al final se desarrollará una simple aplicación demostrando las virtudes de la librería que sea más versátil y fácil de implementar.

Los dispositivos Smartphone actuales tienen suficiente potencia de cálculo para ejecutar correctamente las aplicaciones basadas en los sistemas AR. Aún así, las nuevas tecnologías basadas en el reconocimiento visual del mundo real son mucho más complejas a la hora de crear una realidad virtual paralela, lo cual acaba imposibilitando su óptima ejecución a día de hoy en los dispositivos Smartphone.

-
- E-mail de contacte: peregarciasanahuja@gmail.com
 - Menció: Computació
 - Treball tutoritzat per: Felipe Lumbreras
 - Curs 2014/15

Actualmente las nuevas tecnologías de visión por computador permiten ir más allá a la hora de generar realidades virtuales lo en un futuro permitirá reemplazar todo el concepto actual de realidad aumentada por un nuevo concepto capaz de generar una realidad virtual analizando la realidad física.

2 ESTADO DEL ARTE

Actualmente la tecnología de realidad aumentada está basada en patrones de reconocimiento utilizando una base de datos de imágenes, compuesta por éstas y sus distintas características. Esta tecnología se puede encontrar para distintas plataformas y su uso puede ser mediante licencia comercial o código de dominio público GPL.

Esta tecnología actual se está quedando obsoleta poco a poco ya que la llegada de las nuevas tecnologías de visión por computador permiten utilizar el terreno de la realidad física como soporte para generar la realidad virtual, es decir, eliminar el uso de patrones reales para implementar la realidad virtual en cualquier lugar, sin importar el escenario.

Las tecnologías actuales vSLAM y PTAM son el futuro referente para la realidad aumentada ya que su sistema es el mencionado anteriormente.

Ambas pertenecen a la rama de Visión por Computador y provienen del algoritmo SLAM que permite rastrear el entorno real y generar una realidad virtual paralela sin necesidad de elementos “objetivo” o “targets” ubicados en el entorno real.

- vSLAM (visual Simultaneous Localization And Mapping) es un algoritmo que detecta la realidad física captada a través de un dispositivo de captura de imágenes y la usa para generar un entorno tridimensional (realidad virtual) para uso propio. Se basa en elementos del entorno para generar la realidad virtual. Este algoritmo está pensado y enfocado directamente al estudio de la robótica y sus aplicaciones.
- PTAM (Parallel Tracking And Mapping) es un software de visión por computador que permite rastrear el entorno real (realidad física) mediante un dispositivo de captura de imágenes y generar, en tiempo real, la realidad virtual paralela basándose en las imágenes capturadas. Su funcionamiento es utilizar el entorno como centro de la escena y el dispositivo captador de imágenes se ubica dentro de esa (Posición tridimensional). Así pues cada vez que se va capturando más imágenes se va construyendo la escena (entorno virtual).

Las librerías que implementan la tecnología de realidad aumentada utilizadas en este proyecto son las que permiten utilizar la plataforma ANDROID para desarrollar el sistema. Éstas son VUFORIA, METAIO y AndAR.

2.1 SOFTWARE COMERCIAL

El software comercial corresponde a la librería VUFORIA y METAIO. Ambas son librerías que tienen muchas plataformas de destino y que incluyen características muy interesantes, además se permite su uso para demostraciones o aplicaciones personales.

- METAIO permite generar aplicaciones utilizando su sistema, pero no permite su uso comercial. Aparte, la versión sin licencia incorpora un pequeño logotipo de la librería en la pantalla del dispositivo. Es un sistema muy completo a nivel de usabilidad y tiene muchos usos para la realidad aumentada. Además incorpora una herramienta muy interesante para utilizar las caras de las personas como patrón de reconocimiento.
- VUFORIA permite generar contenido sin ninguna limitación, pero no permite distribuir las aplicaciones que utilizan su sistema sin tener una licencia. Tiene una herramienta muy interesante, basada claramente en el algoritmo SLAM, que permite generar un entorno tridimensional basándose en la realidad física utilizando elementos geométricos como referencia para el entorno generado.

Ambas librerías permiten generar contenido para uso personal, con ciertas limitaciones, pero suficiente como para comprobar su potencial y el funcionamiento de éstas.

Por último, mencionar que ambas siguen el camino del algoritmo SLAM lo que en un futuro cercano permitirá su implementación.

2.2 SOFTWARE NO COMERCIAL

El software no comercial utilizado en este proyecto es AndAR. Se trata de una librería de dominio público que permite generar la realidad aumentada.

Ésta lleva desde el año 2010 sin actualizarse y está en una fase de desarrollo muy temprana. Sólo permite trabajar con patrones bicolor (blanco y negro) parecidos a los códigos QR, ya que no incorpora la posibilidad de gestionar las distintas características de una imagen o patrón en color.

La utilización de ésta librería es bastante desaconsejable ya que su sistema está obsoleto, sobretodo por su uso focalizado en el sistema ANDROID, el cual evoluciona cada poco tiempo y deja los sistemas antiguos sin funcionamiento.

3 METODOLOGÍA

Este trabajo consiste en experimentar con las distintas librerías actuales de realidad aumentada que soportan la plataforma ANDROID, ya que el sistema es gratuito.

Para el estudio de las librerías se generarán unas demostraciones técnicas y se analizarán posteriormente, dando

como resultado, la utilización de una de las librerías para generar una aplicación final.

La aplicación final tendrá elementos básicos de jugabilidad que permitirán al usuario interactuar con la escena generada por el dispositivo. Una vez generada se podrá comprobar como interactúan los elementos de la realidad virtual, los elementos de la realidad física y el usuario, dando así la posibilidad de convertir el proyecto en un videojuego.

Las demostraciones técnicas de las librerías de realidad aumentada METAIO y VUFORIA están desarrolladas bajo el IDE UNITY 4.5 mientras que para la librería AndAR se ha utilizado el IDE ECLIPSE KEPLER.

Las demostraciones servirán para tener una idea de cómo funciona el sistema de realidad aumentada y como afectan los distintos factores ambientales durante la captura de la realidad física.

3.1 OBJETIVOS

- Desarrollar una aplicación para Smartphone con sistema operativo Android (4.2.2)
- Permitir reproducir contenido 3D mediante software de realidad aumentada
- Permita utilizar los componentes del Smartphone
- Muestre elementos básicos para una demostración de una aplicación/juego
- Estudiar las nuevas tecnologías actualmente en fase de desarrollo
- Estudiar las distintas librerías gratuitas disponibles para Android
- Desarrollar una demostración técnica utilizando las librerías gratuitas actualmente disponibles

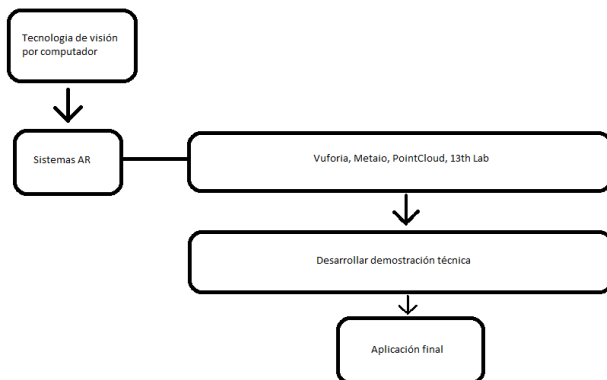


Fig. 1. Esquema de los objetivos

3.2 FUNCIONALIDADES

- La aplicación debe reproducir contenido en 3D.
- La librería utilizada debe ser compatible con UNITY para resolver el código generador de la vista en 3D.
- Estudiar las distintas librerías debe permitir extraer una conclusión técnica basada en el Smartphone utilizado para las pruebas.
- Las demostraciones mostrarán el mismo contenido para poder extraer conclusiones técnicas de

rendimiento.

- La aplicación debe permitir utilizar un “target” plano de reconocimiento o en forma geométrica de cilindro.
- La aplicación final debe usar la librería más versátil y fiable de las estudiadas.
- La aplicación final debe permitir interactuar con el escenario virtual.

3.3 ENTORNO

Para desarrollar las demostraciones se ha utilizado JAVA como lenguaje de programación y el entorno de desarrollo eclipse (KEPLER) y UNITY (4.5).

Las librerías son para Unity, pero una vez generado el proyecto con ésta, se pueden programar los scripts directamente en el IDE Eclipse.

Así pues, el trabajo queda dividido en cuatro partes:

- Librería a estudiar (código JAVA para Android)
- Target Manager de la librería
- UNITY IDE versión 4.5
- ECLIPSE KEPLER IDE con ADK

Adicionalmente, se debe añadir que durante el desarrollo de éste trabajo, el IDE UNITY se ha actualizado a la versión 5.0 y la librería Vuforia, también se ha actualizado a la versión 4.0 BETA.

3.4 PLANIFICACIÓN

La planificación de las semanas será la siguiente:

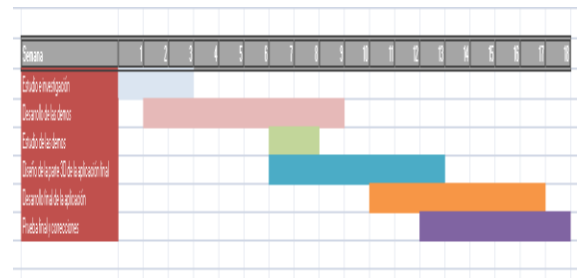


Fig. 2. Planificación de las semanas en format Gantt

Etapa	Semanas
Estudio e investigación	2 Semanas
Desarrollo de las demos	4 Semanas
Estudio de las demos	1 Semanas
Diseño de la parte 3D de la aplicación final	4 Semanas
Desarrollo final de la aplicación	4 Semanas
Prueba final y correcciones	1 Semanas

Cuadro. 1. Planificación del proyecto

4 PROYECTO

El proyecto se divide en 3 partes:

- Desarrollo de las demostraciones
- Analizar las demostraciones
- Desarrollo de la aplicación final

4.1 DEMOSTRACIONES

Para realizar las demostraciones se ha utilizado directamente UNITY 4.5 y su lenguaje de programación CSharp a excepción de cuando se utilizaba la librería AndAR.

El material de diseño (assets) proviene de la tienda online UNITY ya que tiene bastantes diseños gratuitos que cumplen su cometido para este proyecto.

Todo el trabajo realizado sobre las demostraciones y sobre el desarrollo de la aplicación final está definido en cada objeto de la escena. Estos se componen de diferentes “scripts” que definen su comportamiento y sus funciones. Cada vez que la escena se renderiza, todos los objetos del frame anterior son invocados y llevados al estado deseado ejecutando la función principal de actualización (“void update()”) de cada “script”.

Para cada librería se han generado demostraciones basadas en dos escenas. Una de ellas muestra la famosa tetera sin ninguna textura y la otra muestra un cilindro sin textura junto a una esfera con textura de pelota de fútbol.

Además de las escenas, las demostraciones se han desarrollado para comparar las opciones de realidad aumentada de las cuales dispone cada una de las tres librerías comentadas.

Los tres modelos de patrones que se compararán son:

- “Image target”: Un patrón plano representado por una imagen en la realidad física y por un conjunto de características en la realidad virtual.
- “Multi target”: Un cubo tridimensional que tiene una textura en cada uno de los lados. La realidad virtual lo identifica por su forma cuadrangular, por sus texturas adyacentes las unas con las otras y por las características de cada uno de los lados.
- “Cylinder target”: Un cilindro tridimensional recubierto con una textura. La realidad virtual interpreta la continuidad de la textura y las medidas de ésta.

Una vez comentados los tres elementos a utilizar, definiremos para cada librería las opciones disponibles y cómo utilizarlas.

- METAIO: Esta librería soporta los tres modelos de patrones perfectamente. Para el primero y el segundo se ha utilizado la escena de la tetera como ejemplo y para el tercero se ha utilizado la escena de la pelota para detectar el cilindro. Es importante tener en cuenta que la librería utilizada corresponde a la versión gratuita.
- VUFORIA: Al igual que la librería METAIO, ésta soporta los tres modelos y se han generado las

mismas escenas para ellos.

La versión utilizada corresponde a la gratuita, pero la diferencia con la de pago resta en la posibilidad de distribuir comercialmente la aplicación.

- AndAR: Aquí la cosa cambia, sólo se ha podido generar la escena de la tetera con un “image target” generado a partir de una imagen en color blanco y negro, parecida a un código QR, ya que esta librería detecta los cambios de cromatismo en la realidad física, por lo que la manera más fácil de trabajar con ella es utilizando un “image target” bicolor.

Una vez generadas las demostraciones, se analizarán para comprobar cuál es la más indicada para seguir con el proyecto.

4.2 ANÁLISIS DE LAS DEMOSTRACIONES

Los análisis de las demostraciones se centrarán en analizar las librerías de realidad aumentada. Una vez analizadas, se valorará la que más prestaciones y usos tiene y se utilizará para desarrollar la aplicación final.

Cada librería ha sido analizada según elementos de la escena y renderizado así como opciones y usos que se le puede dar.

Debo señalar que en ningún momento se han añadido elementos movibles ni sonidos a la escena, ya que puede provocar un menor rendimiento, pero sí se ha introducido un segundo elemento cilíndrico en las escenas de las librerías VUFORIA y METAIO, para comparar el rendimiento entre ambas.

El dispositivo utilizado para ejecutar las demostraciones ha sido un BQ AQUARIS 5 que tiene un procesador Quad Core Cortex A7 hasta 1,2 GHz, una GPU PowerVR™ SGX544 hasta 286 MHz y 1 GB de RAM.

- VUFORIA: La librería Vuforia es la que menos problemas ha dado a la hora de configurar las opciones de visión del dispositivo de captura de imágenes ya que queda todo muy bien recogido en el archivo de configuración del proyecto. Las opciones que permite están relacionadas con el campo y profundidad de visión del dispositivo así como el “target” principal a definir como punto de partida para generar el escenario tridimensional.

Es completamente gratuita para uso personal y no contiene ninguna marca de su uso a simple vista.

El rendimiento ha sido el mismo usando una imagen bicolor o una con textura. Además, añadir un objeto cilíndrico más a posteriori no provoca ninguna bajada de framerate en la escena. Se mantiene a 30 fps en todo momento.

La detección del “target” es bastante buena, una vez capturado, se puede deformar moviendo el dispositivo. Además el tiempo que tarda en desaparecer la escena en pantalla cuando el “target” no aparece es de 1 segundo.

Finalmente, añadir que esta librería tiene una opción de configuración muy interesante llamada “Extender tracking” que una vez generada la escena, si ésta sobresale de la pantalla del dispositivo (viewport), te permite incrustar el escenario en pantalla para poder visualizarlo correctamente utilizando los sensores de giroscopio.

- **METAIO:** La librería Metaio es muy parecida a Vuforia, pero con una marca de agua para la versión “gratuita”.

La configuración viene recogida en parte en un archivo de configuración, pero hay ciertas opciones de visualización que si no se codifican directamente en el fichero fuente, no aparecen.

El rendimiento utilizando los mismos “targets” es de 30 fps sin ningún tipo de ralentización al añadir un segundo elemento cilíndrico y permite girar alrededor del “target” y deformarlo bastante hasta perder la escena. En general la librería de AR de Metaio es muy completa.

Como añadido principal, Metaio permite la detección de caras de personas (en versión BETA) que, aunque no se haya probado en este trabajo, puede dar muy buenos resultados al uso.

- **AndAR:** La librería andAR utiliza ARToolKit para mostrar un entorno tridimensional basado en un elemento “image target”.

A diferencia de las otras librerías, ésta no permite crear el proyecto en Unity ya que no dispone de módulo propio para generar el contenido posterior, así que se utiliza un entorno de desarrollo para Android (como ECLIPSE) para gestionar el código de la aplicación.

Lo bueno que tiene es que permite cargar objetos tridimensionales, como el barco de la pantalla, alojados en la memoria del dispositivo, por lo que en realidad se asemeja bastante a las otras dos librerías a la hora de añadir elementos 3D.

La escena de la tetera se mantiene en 10~8 frames por segundo y la detección es muy lenta comparada con las otras dos librerías.

Por otra parte, no he conseguido cargar un modelo geométrico ni generar más de un elemento en pantalla ya que daba muchos problemas al mostrar la escena, hasta tal punto que se cerraba la aplicación por la mala gestión de la memoria de la librería. Además tampoco ha sido posible utilizar una imagen con textura, sin saber el porqué, pero no detectaba el patrón.

Como conclusión final de esta librería cabe destacar la facilidad de su uso y su deficiente sistema, pero teniendo en cuenta que la última actualización es del 2010, casi no se puede equiparar con las otras dos.

Los “image targets” utilizados corresponden a una imagen sin textura bicolor y una textura de 24 bits de profundidad de color, lo cual ha influido directamente en

el rendimiento.

4.3 APLICACIÓN FINAL

El resultado obtenido de las demostraciones permite utilizar la librería VUFORIA de QUALCOMM como base para desarrollar una aplicación nativa para Smartphone ANDROID que permita interactuar con los elementos y mostrar una pequeña parte de lo que llegaría a ser un videojuego utilizando los elementos de las demostraciones.

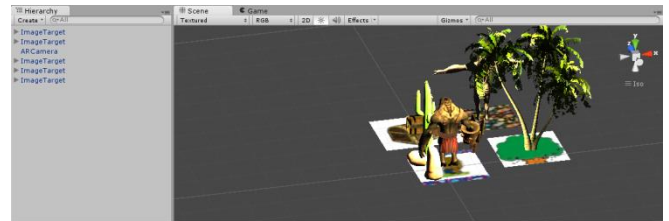


Fig. 3. Escena en UNITY

La demostración se ha generado utilizando el motor gráfico UNITY 4.5 y aprovechando los “assets” de su tienda online, lo que permitió ahorrar tiempo durante la parte de diseño.

La aplicación utiliza 5 “image targets” para generar una pequeña escena que permite a los elementos interactuar entre ellos y el usuario. Los targets se pueden encontrar en el apartado Anexos.

La escena está compuesta por un tablero inicial, para el cual se ha utilizado un modelado tridimensional parecido a un desierto. Aparte también existen los elementos dependientes de los “image targets”.

Estos elementos pueden ser generados sin depender del tablero inicial.

El protagonista principal está representado por un modelado 3D que parece un ogro.

El concepto de la aplicación consiste en poder generar elementos en pantalla mediante los “image targets” e interactuar con ellos mediante la pantalla táctil del dispositivo.

El desarrollo se puede dividir en 5 puntos:

- Implementar la escena con los “image targets” necesarios.
- Implementar todo el contenido tridimensional de la escena.
- Implementar las condiciones que permiten a los elementos realizar la acción correspondiente.
- Utilizar para cada elemento la acción correspondiente.
- Una vez generado el movimiento o efecto, se puede volver a generar el mismo repitiendo la acción sobre la pantalla táctil del dispositivo.

Para realizar el primer punto se requiere definir todos los “image targets” en el target manager de la librería Vuforia y añadirlos al proyecto de Unity. Una vez añadidos los targets, se procede a añadir el contenido tridimensional a

la escena.



Fig. 4. Escena de realidad aumentada

Cuando el diseño de la escena queda terminado, se empieza con la programación de los elementos condicionales, los cuales se encargan de ejecutar las acciones correspondientes.

Éstas se deben añadir a la clase encargada de comprobar si el patrón de alguno de los “image targets” de la escena corresponde con la imagen de la cámara del dispositivo Smartphone.

Una vez añadidas las condiciones, se procede a definir las acciones de los objetos de la escena. Cada acción se define en la clase del objeto y se ejecutará en cada renderizado (frame).

Finalmente se define el control del usuario mediante la pantalla táctil del dispositivo, añadiendo la posibilidad de multiplicar uno de los elementos.

4.4 INFORMACIÓN TÉCNICA DE LA APLICACIÓN FINAL

Elementos que forman la escena:

Nombre	Triangulos
Terreno/Tablero	524
Ogro (Personaje principal)	956
Palmera	349
Zombie	1492
Cofre	208
Cactus y piedras	127

Cuadro. 2. Elementos de la escena

Como se puede apreciar, con el hardware utilizado para realizar el proyecto ha habido problemas si se juntan varios elementos a la vez en la misma escena, los cuales derivan en posicionamiento incorrecto de los elementos en la escena o reinicios constantes de las acciones, lo que al final, imposibilita el correcto desarrollo de la aplicación.

Nº Elems.	Triangulos	Frames (fps)	Estabilidad
1 (tablero)	524	30	Óptima
1 + tablero	1480	30	Óptima
2 + tablero	1829	25	Optima
3 + tablero	3321	19	Sin interferencias
4 + tablero	3529	12	Errores en target
5 + tablero	3656	7	Sync Error

Cuadro. 3. Comparativa técnica

5 RESULTADOS

Las demostraciones se han utilizado para definir un camino a seguir para desarrollar la aplicación final, pero las mismas sirven para ver las diferencias entre las distintas librerías.

Cada librería tiene sus propias características, pero las librerías METAIO y la VUFORIA muestran un rendimiento óptimo, así como la facilidad para integrar el proyecto en distintos entornos de desarrollo.

Ambas proporcionan herramientas muy útiles para generar contenido de realidad aumentada, ya que su funcionamiento es muy simple y teniendo unos conocimientos del entorno donde se quiera usar, se pueden generar muchas aplicaciones.

El resultado del proyecto se puede definir como el principio del desarrollo de un videojuego de realidad aumentada, ya que la aplicación final permite interactuar con la realidad virtual generada a partir de los patrones descritos en la realidad física.

6 CONCLUSIÓN

Visto el estudio y las demostraciones técnicas de las distintas librerías, el proyecto se ha centrado en utilizar la librería Vuforia para diseñar e implementar una aplicación que permite ver cómo se desarrolla un proyecto basado en realidad aumentada.

El procedimiento del desarrollo es básico para optimizar la aplicación, ya que como se puede comprobar con los datos técnicos, todavía le queda un tiempo de desarrollo a ésta librería.

Actualmente los smartphones tienen suficiente capacidad de procesamiento gráfico para recrear una escena tridimensional, pero la carga que supone la librería al tener muchos elementos dependientes de patrones, no permite gestionar correctamente los recursos y no renderiza la escena de manera óptima.

Personalmente creo que VUFORIA es una librería de realidad aumentada muy a tener en cuenta, ya que aparte de las ventajas comentadas anteriormente, tiene un rendimiento bastante fluido. Además su manejo es muy sencillo, incluso para diseñadores de modelado tridimen-

sional, que no quieren pararse a programar ningún elemento para poder renderizar simples pruebas.

Ahora bien, en este proyecto no se ha utilizado METAIO por una simple razón, si no se dispone de la licencia, no se permite configurar la librería.

Como futuro trabajo se debería adquirir la licencia de la librería METAIO y realizar comparaciones con VUFORIA, ya que en el momento de terminar este proyecto, ambas librerías han evolucionado y presentan elementos muy interesantes.

Como conclusión personal, el proyecto me ha sido de gran ayuda para adentrarme en el mundo de los videojuegos en plataformas móviles. Además he aprendido a programar y manejar el IDE UNITY junto con el “kit nativo de desarrollo de ANDROID” (NDK).

AGRADECIMIENTOS

Agradezco a mi tutor del proyecto la oportunidad de estudiar esta tecnología en un proyecto final ya que me ha dado muchas ideas para desarrollar futuros proyectos.

A mi familia por ayudarme a realizar las pruebas y recabar los datos.

También quiero agradecer a la comunidad de desarrolladores de UNITY que me han dado soporte online.

BIBLIOGRAFIA

- [1] 13th Lab (<http://13thlab.com/>), 2015.
- [2] PTAM Parallel Tracking And Mapping, <http://www.robots.ox.ac.uk/~gk/PTAM/>. 2014.
- [3] AR ToolKit, <http://artoolkit.sourceforge.net/>. 2014.
- [4] Metaio, <http://dev.metaio.com/creator/>. 2014.
- [5] PointCloud SDK, <http://developer.pointcloud.io/sdkdocs/index.html>. 2014.
- [6] AndAR- Android Augmented Reality, <https://code.google.com/p/andar/>. 2010
- [7] VUFORIA fórum, <https://developer.vuforia.com/forum/android/>. 2014
- [8] UNITY fórum, <http://forum.unity3d.com/>. 2014

8 ANEXO

Imágenes de los targets



Fig. 5. Target: Tablero/Terreno



Fig. 6. Target: Duende



Fig. 7. Target: Cofre



Fig. 8. Target: Palmera



Fig. 9. Target: Personaje principal Ogro

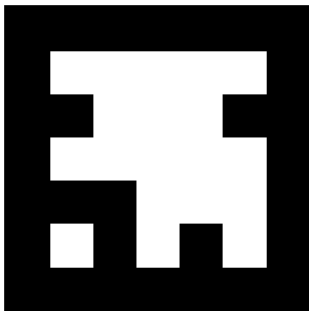


Fig. 10. Target: Target (FrameMarker)



Fig. 11. Target: Ogro